REMARKS

This is filed in response to the Office Action dated September 4, 2008, citing objection to claims 32 – 34 and 47 – 56 and rejecting claims 1 – 63 under 35 U.S.C. § 103. In view of the amendments above and the remarks that follow, the Applicants submit that all pending claims are in condition for allowance.

I. Objections

The Examiner cites objection to claims 32 – 34 and 47 – 56. The Applicants amend claims 32 and 47 to address the objection.

II. Rejection of Claims 1 – 63 Under 35 U.S.C. § 103

A. Claims 1 – 10, 13 – 19, 23 – 43, 46 – 49 and 52 – 63

Claims 1 – 63 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Emer et al. (US 6,493,741; hereinafter "Emer") in view of Kelsey et al. (US 7,082,519; hereinafter "Kelsey"), which incorporates by reference Eggers (Eggers et al. "Simultaneous Multithreading: A Platform for Next-Generation Processors, IEEE, 1997; pages 12 – 19), and Sekiguchi et al. (US 2001/0016879; hereinafter "Sekiguchi").

Claim 1 is directed to an embedded processor comprising a plurality of processing units that each execute processes or threads (collectively, "threads"). One or more execution units are shared by the processing units and execute instructions from the threads. An event delivery mechanism is in communications coupling with the plurality of processing units and delivers events (e.g., interrupts) to respective threads without execution of instructions by the processing units.

Neither Emer, Kelsey nor Sekiguchi, individually and in combination, teach or suggest an embedded processor meeting the limitations of claim 1. More specifically, Emer purports to disclose a method for resolving the problem of spin-lock in an SMT architecture by halting, or

"quiescing," spin-locking threads while they are waiting for some event, i.e., the availability of a lock. *See* Emer, col. 3, lines 24 – 27. Nowhere does that publication teach or suggest, *inter alia*, an event delivery mechanism that is in communication with a plurality of processing units and that delivers events to respective threads without execution of instruction by the processing units, e.g., as required by claim 1. The Office Action does not contend otherwise — but, instead, asserts that Kelsey and Sekiguchi remedy this deficiency.

More specifically, the Office Action contends that Kelsey teaches an event delivery mechanism capable of delivering events to threads with which they are associated. *See*, Office Action, mailed September 4, 2008, page 4. The Office Action does *not* contend — nor does Kelsey actually teach or suggest — that such event delivery can be accomplished without execution of instructions by the processing units, e.g., as required by claim 1. For this, the Office Action relies on Sekiguchi.[1]

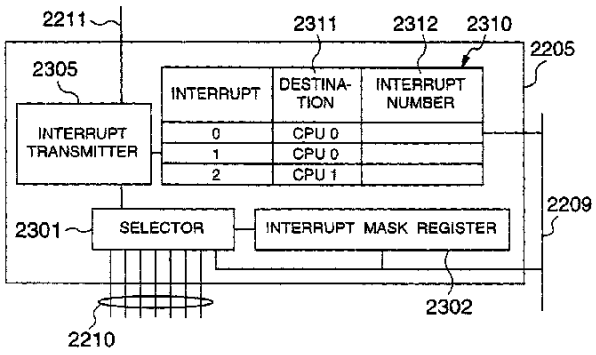
That publication purports to disclose a method and system for configuring a plurality of operating systems. *See* Sekiguchi, ¶ 0015. However, it too fails to teach or suggest, *inter alia,* an event delivery mechanism capable of delivering events to threads without execution of instructions by the processing units, e.g., as required by pending claim 1.

The Office Action incorrectly asserts that Sekiguchi "teaches an event delivery mechanism that delivers events to threads with which those events are associated, wherein the event delivery mechanism delivers each such event to the respective thread without execution of instructions by said processing units." *See* Office Action, mailed September 4, 2008, page 4. However, the Office Action unfairly attributes to the Sekiguchi interrupt controller capabilities that simply are not disclosed in that publication.

[1] The Applicants do not concede that Kelsey teaches even that which the Office Action contends – i.e., an event delivery mechanism capable of delivering events to threads with which they are associated. The teachings of Kelsey in this particular regard are irrelevant since, as discussed below, Sekiguchi fails to remedy the deficiencies of Kelsey implicitly admitted in the Office Action.

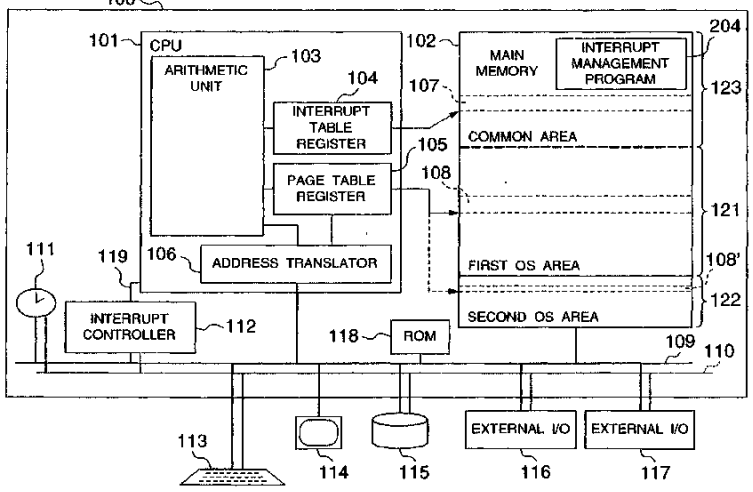
Specifically, for example, nowhere does Sekiguchi teach or suggest that the interrupt controller can deliver events to threads. At most, that publication suggests that the controller can specify a CPU destination for an interrupt — not a particular thread, process or otherwise! For example, Figure 23 of Sekiguchi, reprinted left, clearly shows that the only destination specified for an interrupt is a CPU. There is no indication of any capability for delivery to a thread, process, or otherwise.

FIG. 23



Indeed, closer scrutiny of Sekiguchi reveals that processor instructions are actually required for interrupt delivery. Referring to Figure 1 of that publication, reprinted below, there is shown, among other things, an interrupt table register 104 within the CPU 101. In regard to that drawing, Sekiguchi says "[w]hen an interrupt occurs, the processor 101 receives the interrupt number from the interrupt controller 112. By using this number as a search index, the processor acquires an interrupt handler address from the interrupt table 107 to pass the control to the interrupt handler." *See* Sekiguchi, ¶ 0050, lines 10 – 15.

FIG. 1



It is thus quite clear that Sekiguchi requires processor instructions to handle interrupts. This is contrary to the recitation of claim 1, which calls for delivery of events to respective threads without execution of instructions by the processing units.

In view of the foregoing, it is evident that the combination of Emer, Kelsey and Sekiguchi fails to teach, suggest or otherwise render unpatentable the subject matter of claim 1. The same is true for claims 2 – 6 which depend from claim 1 and recite further limitations thereon.

For like reasons, Emer, Kelsey and Sekiguchi fail to teach, suggest or otherwise render unpatentable the subject matter of claims 7, 17, 28, 32, 35, 41, 47, 57 and 61, which parallel claim 1 in regards relevant to the arguments above. The remaining claims depend from either claim 7, 17, 28, 32, 35, 41, 47, 57 or 61, reciting further limitations thereon. For at least the reasons above, they too are patentable over Emer, Kelsey and Sekiguchi.

B. Claims 11, 12, 20 – 22, 44, 45, 50 and 51

Claims 11, 12, 20 – 22, 44, 45, 50 and 51 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Emer in view of Kelsey, which incorporates Eggers, and Sekiguchi, and further in view of Microsoft Computer Dictionary (Fifth Edition, Microsoft Press, 2002). These claims depend from claims 7, 17, 41 and 47, and are patentably distinct from the teachings of Emer in view of Kelsey and Sekiguchi for at least the reasons discussed above in connection with claim 1. Insofar as Microsoft Computer Dictionary fails to remedy the deficiencies of those other references — namely, failing to teach or suggest *inter alia* an event delivery mechanism that is in communication coupling with a plurality of (virtual) processing units and that delivers events to respective threads with which those events are associated without execution of instruction by said processing units — claims 11, 12, 20 – 22, 44, 45, 50 and 51 are patentably distinct from the combination of Emer, Kelsey and Microsoft Computer Dictionary.

C. Claims 26 and 55

Claims 26 and 55 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Emer in view of Kelsey, which incorporates Eggers, and Sekiguchi, and further in view of

Blandy (US 6,912,647). These claims depend from claims 17 and 47, respectively, and are patentably distinct from the teachings of Emer in view of Kelsey and Sekiguchi for at least the reasons above. Insofar as Blandy fails to remedy the deficiencies of those other references — namely, failing to teach or suggest *inter alia* an event delivery mechanism that is in communication coupling with a plurality of (virtual) processing units and that delivers events to respective threads with which those events are associated without execution of instruction by said processing units — claims 26 and 55 are patentably distinct from the combination of Emer, Kelsey and Blandy.

V. Conclusion

In view of the foregoing, the Applicants believe that the application is in condition for allowance. The Examiner is encouraged to telephone the undersigned attorney for Applicants if such communication will expedite prosecution of this application.

Respectfully submitted,

Date: January 5, 2009

/Benjamin E. Berman/

Benjamin E. Berman (U.S. Reg. 61,507)
Attorney for Applicants
Nutter McClennen & Fish LLP
World Trade Center West
155 Seaport Boulevard
Boston, MA 02210-2604
Tel: (617) 439-2000
Fax: (617) 310-9000